UM–SJTU Joint Institute

Introduction to Operating Systems
(VE482)

Lab 8

Linux kernel

Pgroup-pair 6

Name	ID
Xie Jinglei	517370910022
Chen Yaxin	517021911020

Date: 23 Nov. 2019

1 Concepts

1.1 Kernel modules

1. What is a kernel module, and how does it different from a regular library?

   Kernel modules are pieces of code that can be loaded and unloaded into the kernel upon demand. They extend the functionality of the kernel without the need to reboot the system.

   Main differences from library:

   – Kernel modules can be dynamically loaded, so that you can add functionality to the kernel while the system is up and running.

   – kernel modules are linked only to the kernel, which means they do not link in the same libraries that user programs link in. The only functions a kernel module can call are functions that are exported by the kernel.

2. How to compile a kernel module?

   A simple example:

```
obj-m := hello-1.o

all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

   obj-m: specifies object files which are built as loadable kernel modules.
   uname -r: get name and information about current kernel.
   -C dir: change to directory **dir** before reading the makefiles.
   M=$(PWD): a shell variable that stores the current source path to your kernel module. **make** needs to store this as it switches to the kernel build path.

1.2 Other related questions

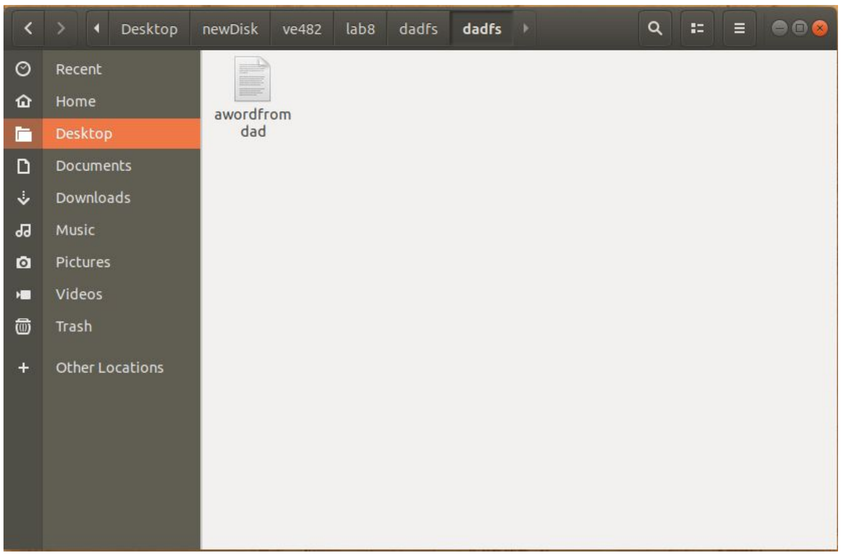
1. How are mutex defined and used? How good is this approach?
   static DEFINE_MUTEX() is used. **DEFINE_MUTEX** defines and initializes mutex at the same time. It is useful for defining global mutexes.
2. How is information shared between the kernel and user spaces?
   The old version uses the functions **copy_to_user** and **copy_from_user**. To adapt to the new kernel, we use **copy_to_iter** and **copy_from_iter**. Note that the input arguments of the functions are different.

1.3 Documentation of changes

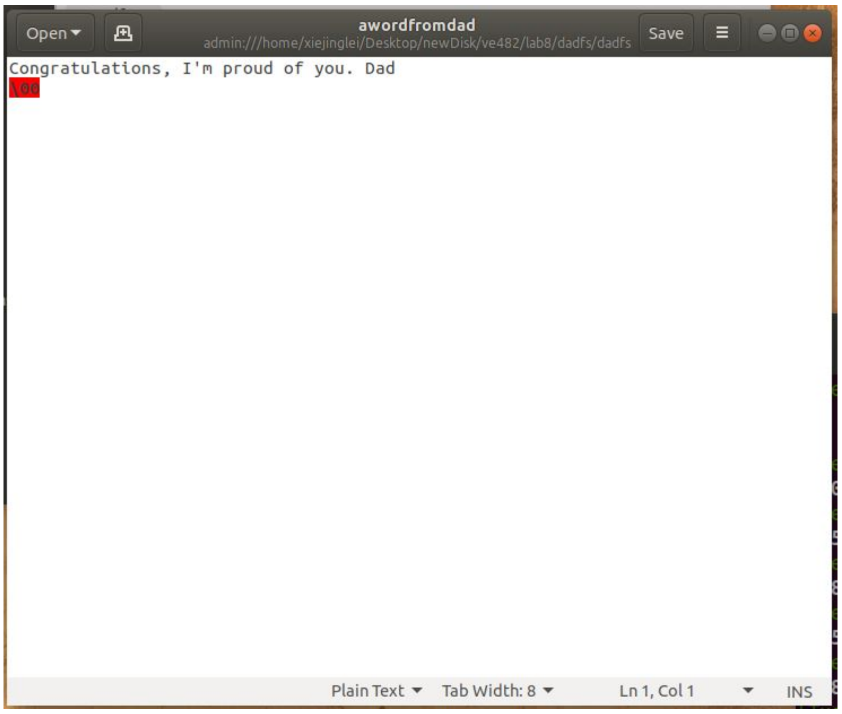
Please see README.

2 Results

After we mount the disk, a file "awordfromdad" appeared in the original fs:

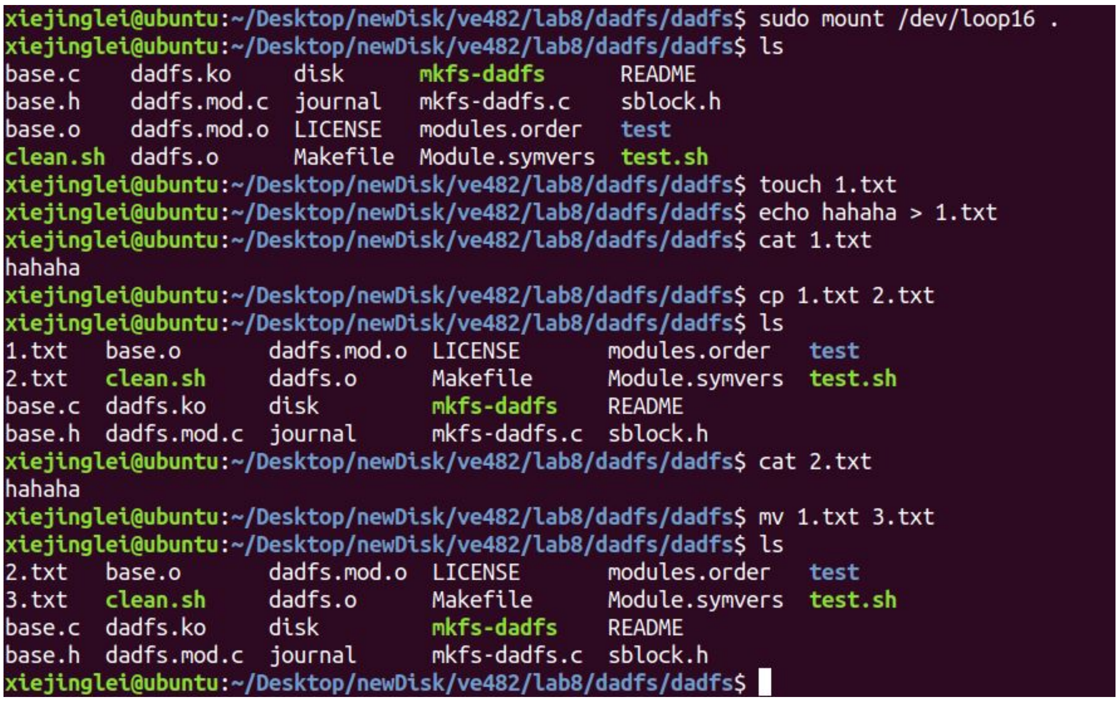


which says...


Open
awordfromdad
admin:///home/xiejinglei/Desktop/newDisk/ve482/lab8/dadfs/dadfs
Save
Plain Text
Tab Width: 8
Ln 1, Col 1
INS

```
Congratulations, I'm proud of you. Dad
\00
```

For testing, normal commands seems to work well,



```
xiejinglei@ubuntu:~/Desktop/newDisk/ve482/lab8/dadfs/dadfs$ sudo mount /dev/loop16 .
xiejinglei@ubuntu:~/Desktop/newDisk/ve482/lab8/dadfs/dadfs$ ls
base.c    dadfs.ko     disk      mkfs-dadfs      README
base.h    dadfs.mod.c  journal   mkfs-dadfs.c    sblock.h
base.o    dadfs.mod.o  LICENSE   modules.order   test
clean.sh  dadfs.o      Makefile  Module.symvers  test.sh
xiejinglei@ubuntu:~/Desktop/newDisk/ve482/lab8/dadfs/dadfs$ touch 1.txt
xiejinglei@ubuntu:~/Desktop/newDisk/ve482/lab8/dadfs/dadfs$ echo hahaha > 1.txt
xiejinglei@ubuntu:~/Desktop/newDisk/ve482/lab8/dadfs/dadfs$ cat 1.txt
hahaha
xiejinglei@ubuntu:~/Desktop/newDisk/ve482/lab8/dadfs/dadfs$ cp 1.txt 2.txt
xiejinglei@ubuntu:~/Desktop/newDisk/ve482/lab8/dadfs/dadfs$ ls
1.txt   base.o       dadfs.mod.o  LICENSE       modules.order   test
2.txt   clean.sh     dadfs.o      Makefile      Module.symvers  test.sh
base.c  dadfs.ko     disk         mkfs-dadfs    README
base.h  dadfs.mod.c  journal      mkfs-dadfs.c  sblock.h
xiejinglei@ubuntu:~/Desktop/newDisk/ve482/lab8/dadfs/dadfs$ cat 2.txt
hahaha
xiejinglei@ubuntu:~/Desktop/newDisk/ve482/lab8/dadfs/dadfs$ mv 1.txt 3.txt
xiejinglei@ubuntu:~/Desktop/newDisk/ve482/lab8/dadfs/dadfs$ ls
2.txt   base.o       dadfs.mod.o  LICENSE       modules.order   test
3.txt   clean.sh     dadfs.o      Makefile      Module.symvers  test.sh
base.c  dadfs.ko     disk         mkfs-dadfs    README
base.h  dadfs.mod.c  journal      mkfs-dadfs.c  sblock.h
xiejinglei@ubuntu:~/Desktop/newDisk/ve482/lab8/dadfs/dadfs$
```

Perhaps we are having some trouble with “cd”, if we try to cd into the parent directory, like “cd ..”. It gives “permission denied”.

References

[1] https://wiki.archlinux.org/index.php/Kernel_module

[2] https://stackoverflow.com/questions/37507320/how-to-compile-a-kernel-module

[3] https://elixir.bootlin.com/linux/latest/source/include/linux/fs.h#L1789